

## REMARKS

Applicant is in receipt of the Office Action mailed October 2, 2002. Claims 1-36 are pending in the application. Claims 12-27 were allowed. Claims 1-11 and 28-36 were rejected under §101. Claims 28 was also rejected for double patenting.

### §101 Rejections

Claims 1-11 and 28-36 were rejected under §101 because they were directed toward a “computer program”. Applicant has amended each of these claims whereby the claims are now directed toward “an electronic medium comprising a computer program”. Thus, Applicant submits that this rejection has been overcome.

### Double Patenting Rejection

Claims 28 was rejected for double patenting because it conflicted with claim 1 of Application No. 09/760,511. Applicant is not sure from the Office Action whether the double patenting rejection is a same type double patenting rejection or an obviousness type double patenting rejection. Applicant has reviewed claim 28 of the present application and claim 1 of Application Serial No. 09/760,511. Applicant submits that the subject matter of claim 28 of the present application is quite different from, and both novel and non-obvious in view of, claim 1 of Application Serial No. 09/760,511. For example, claim 28 recites in part:

“Creating a plurality of data structures and threads, wherein each data structure corresponds to a particular type of object in the hierarchy, wherein each object in the hierarchy has an entry in at least one of the data structures, and wherein each thread is configured to execute one of the data structures”

In contrast, claim 1 of Application No. 09/760,511 recites in part:

“Create a plurality of data structures and threads, wherein the threads are configured to generate messages to specify state changes,”

Applicant submits that the above underlined limitation is not found in claim 28 of the present application. Thus, Applicant submits that a same type double patenting rejection is not appropriate. At most, an obviousness type double patenting rejection could be made. Applicant submits herewith a terminal disclaimer to obviate any obviousness type double patenting rejection. Thus, Applicant submits that claims 28 and those dependent thereon are allowable.

## CONCLUSION

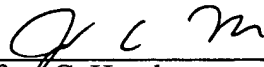
Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Conley, Rose, & Tayon, P.C. Deposit Account No. 50-1505/5181-45501/JCH.

Also enclosed herewith are the following items:

- ☒ Return Receipt Postcard
- ☒ Terminal Disclaimer
- ☒ Fee Authorization

Respectfully submitted,



---

Jeffrey C. Hood  
Reg. No. 35,198  
ATTORNEY FOR APPLICANT(S)

Conley, Rose & Tayon, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
Phone: (512) 476-1400  
Date: 12/70/2002

Amendments to claims with changes highlighted

1. (Amended) [A computer program embodied on an] An electronic medium comprising a computer program, wherein the computer program is configured to:

create a scene graph by:

adding one or more objects to a hierarchy by defining parent-child relationships for the one or more objects;

generate a parallel representation of the scene graph by:

determining whether a data structure corresponding to the one or more object types already exist;

if a corresponding data structure does not already exist, create a new data structure corresponding to the type of object;

if a corresponding data structure does already exist, create a new entry in the corresponding data structure;

create at least one thread configured to process each data structure; and

render the scene graph by executing at least a subset of the threads.

2. (Amended) The [computer program] electronic medium of claim 1, wherein the data structures provide a parallel context from which to access the objects in the scene graph.

3. (Amended) The [computer program] electronic medium of claim 1, wherein the data structures provide alternate access to data from the objects in the scene graph.

4. (Amended) The [computer program] electronic medium of claim 1, wherein the data structures provide an alternate ordering of information in the scene graph.

5. (Amended) The [computer program] electronic medium of claim 1, wherein the entries in the data structures comprise pointers to the data in the scene graph.

6. (Amended) The [computer program] electronic medium of claim 1, wherein at least a subset of the entries in the data structures are copies of the data in the scene graph.

7. (Amended) The [computer program] electronic medium of claim 1, wherein the objects in the scene graph comprise object data, wherein the entries in the data structures comprise pointers to the object data.

8. (Amended) The [computer program] electronic medium of claim 1, wherein the entries in the data structures comprise pointers to objects in the scene graph.

9. (Amended) The [computer program] electronic medium of claim 1, wherein said adding one or more objects to a hierarchy by defining parent-child relationships for the one or more objects comprises inserting a pointer to an object in the hierarchy.

10. (Amended) The [computer program] electronic medium of claim 1, further comprising generating one or more threads for each data structure, wherein the threads operate on the data structures.

11. (Amended) The [computer program] electronic medium of claim 1, further comprising generating one or more threads for each data structure, wherein the threads operate on the data structures.

28. (Amended) [A computer program embodied on an] An electronic medium comprising a computer program, wherein the computer program is configured to:  
create a scene graph by:

adding a first object to a hierarchy;

adding a second object to the hierarchy;

defining a parent-child relationship between the first and second objects;

creating a plurality of data structures and threads, wherein each data structure

corresponds to a particular type of object in the hierarchy, wherein each object in

the hierarchy has an entry in at least one of the data structures, and wherein each thread is configured to execute one of the data structures; and rendering the scene graph by executing the threads in lieu of traversing the scene graph.

29. (Amended) The [computer program] electronic medium of claim 28, wherein each data structure has a corresponding update thread configured to update the data structure.

30. (Amended) The [computer program] electronic medium of claim 28, wherein the scene graph includes behavior data for the objects.

31. (Amended) The [computer program] electronic medium of claim 28, wherein the scene graph includes environmental data.

32. (Amended) The [computer program] electronic medium of claim 28, wherein the scene graph includes lighting and fog information.

33. (Amended) The [computer program] electronic medium of claim 28, wherein the program is configured to optimize the scene graph before creating the plurality of data structures and the threads.

34. (Amended) The [computer program] electronic medium of claim 28, wherein the program is configured to optimize the data structures by stripifying geometry data in the data structure.

35. (Amended) The [computer program] electronic medium of claim 28, wherein the program is configured to optimize the data structures by splitting nodes in the data structures.

36. (Amended) The [computer program] electronic medium of claim 28, wherein the program is configured to optimize the data structures by flattening the scene graph before creating the plurality of data structures and the threads.